Gaussian Processes for Classsification & Brief Summary of the Course

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

November 15, 2017

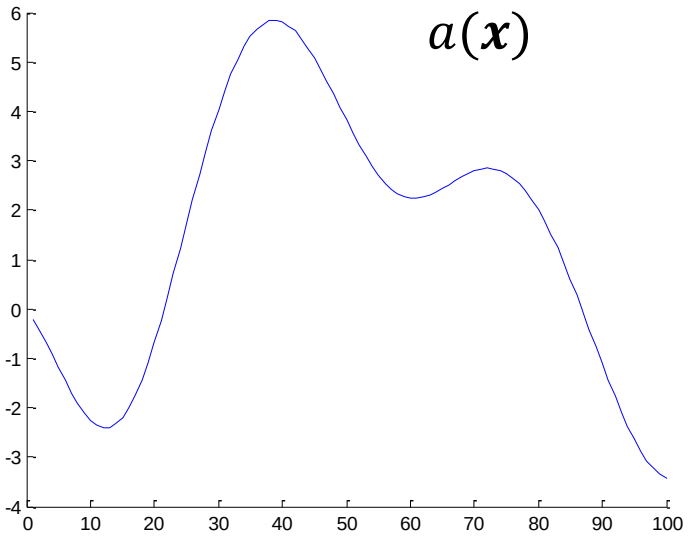
Contents

- Gaussian Process Classification, Laplace Approximation, Connections to Bayesian Neural Networks
- Markov and Chebyshev Inequalities, Law of Large Numbers
- MLE for Gaussians, Robbins-Monro, Curse of Dimensionality
- Bayes' theorem and Gaussian Linear Models, Posterior Inference and Prediction, Marginal Likelihood
- Exponential family of distributions, Conjugate priors
- Empirical Bayes, Evidence approximation, Model Validation and Bayes factors, Occam's Razor
- Laplace approximation
- Sampling from an arbitrary distribution, Inverse Method, Rejection Sampling, Importance Sampling, Gibbs Sampling, MCMC, Metropolis-Hastings, Sequential Importance Sampling and Particle Methods, Reversible Jump MCMC
- Latent Variables and Expectation-Maximization
- Probabilistic PCA and Generative View

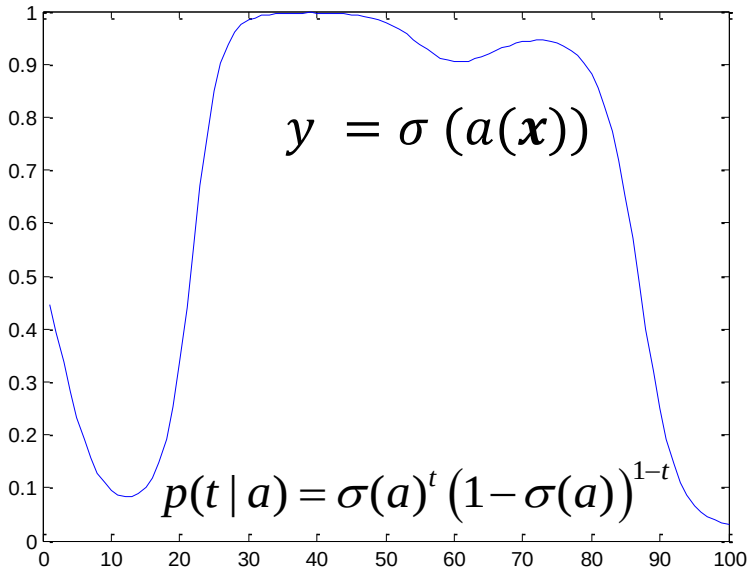
Following closely: Bishop CM, *Pattern Recognition and Machine Learning*, Springer, 2006 (Chapter 6)

Gaussian Processes for Classification

- Objective: model posterior probabilities of the target variable for a new input
- Problem: we need to map values to interval $(0, 1)$
- Solution: use a Gaussian process together with a non-linear activation function (e.g., sigmoid)



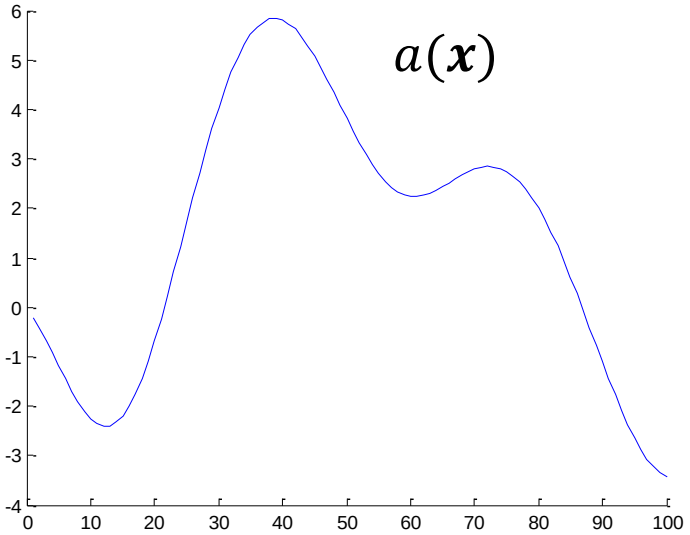
Sample from a Gaussian process over $a(x)$

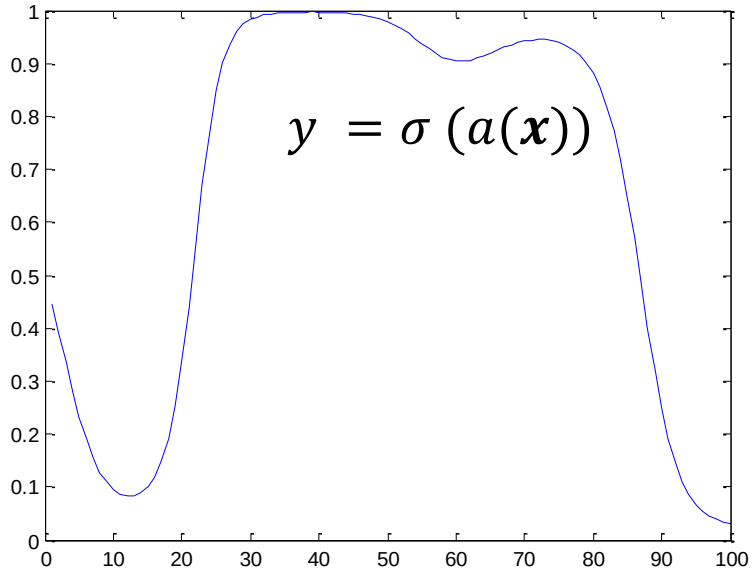


MatLab Code

Gaussian Process for Classification

- Consider a two-class problem with target values t ∈ {0, 1}. Define a Gaussian process over a function $a(\boldsymbol{x})$ and transform a using the logistic sigmoid to $y = \sigma(a(\boldsymbol{x}))$.
- This will give a non-Gaussian stochastic process over functions $y(\boldsymbol{x})$ where y ∈ {0, 1}.





MatLab Code

- The probability over the target t is a Bernoulli: $p(t \mid a) = \sigma(a)^t \left(1 - \sigma(a)\right)^{1-t}$

Gaussian Process for Classification

- We denote the training set inputs by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ with corresponding observed target variables $\boldsymbol{t} = (t_1, \ldots, t_N)^T$.
- We also consider a single test point $\boldsymbol{x}_{N+1}$ with target value t_{N+1}.
- Our goal is to determine the predictive distribution $p(t_{N+1}|\boldsymbol{t})$, where we left the conditioning on the input variables implicit.
- Introduce a Gaussian process prior over the vector $\boldsymbol{a}_{N+1}$, which has components $a(\boldsymbol{x}_1), \ldots, a(x_{N+1})$.
- This defines a non-Gaussian process over $\boldsymbol{t}_{N+1}$, and by conditioning on $\boldsymbol{t}_N$ we obtain the required predictive distribution $p(t_{N+1}|\boldsymbol{t})$.
- The Gaussian process prior for $\boldsymbol{a}_{N+1}$ takes the form

$$p(\boldsymbol{a}_{N+1}) = \mathcal{N}(\boldsymbol{a}_{N+1} \mid 0, \boldsymbol{C}_{N+1})$$

Gaussian Process for Classification

- The covariance matrix does not include a noise term because we assume that all training data points are correctly labeled.

- However, for numerical reasons it is convenient to introduce a noise-like term governed by a parameter ν that ensures that the covariance matrix is positive definite. Thus the covariance matrix $\boldsymbol{C}_{N+1}$ has elements given by

$$\boldsymbol{C}\left(\boldsymbol{x}_n, \boldsymbol{x}_m\right) = k\left(\boldsymbol{x}_n, \boldsymbol{x}_m\right) + \nu\delta_{nm}$$

- The kernel k is positive semidefinite and v is fixed. $k(\boldsymbol{x}, \boldsymbol{x}')$ is governed by parameters θ that we will compute from the training data.

- For two class problems, it is sufficient to predict:

$$p(t_{N+1} = 1 \mid \boldsymbol{t}_N)$$

Gaussian Process for Classification

- The predictive distribution is given as:

$$p(t_{N+1}=1|\boldsymbol{t}_N)=\int p(t_{N+1}=1,a_{N+1}|\boldsymbol{t}_N)\,da_{N+1}$$

$$=\int p(t_{N+1}=1|a_{N+1})\,p(a_{N+1}|\boldsymbol{t}_N)\,da_{N+1}=\int \sigma(a_{N+1})\,p(a_{N+1}|\boldsymbol{t}_N)\,da_{N+1}$$

- The integral is analytically intractable. Can be approximated with sampling (Neal)
- Integration can be analytical if $p(a_{N+1}|t_N)$ can be approximated with a Gaussian (Williams & Barber, Gibbs & MacKay, Gibbs)
- Approximations based on EP can also be used (Opper & Winther, Minka, Seeger)

- Neal, R. M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Department of Computer Statistics, University of Toronto.
- Williams, C. K. I. and D. Barber (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 1342–1351.
- Gibbs, M. N. and D. J. C. MacKay (2000). Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks* **11**, 1458–1464.
- Gibbs, M. N. (1997). *Bayesian Gaussian processes for regression and classification*. Phd thesis, University of Cambridge.
- Opper, M. and O. Winther (2000b). Gaussian processes for classification. *Neural Computation* **12**(11), 2655–2684.
- Minka, T. (2001b). *A family of approximate algorithms for Bayesian inference*. Ph. D. thesis, MIT.
- Seeger, M. (2003). *Bayesian Gaussian Process Models: PAC-Bayesian Generalization Error Bounds and Sparse Approximations*. Ph. D. thesis, University of Edinburg.

Gaussian Process for Classification

- We need to compute $p(t_{N+1}=1|\boldsymbol{t}_N)=\int \sigma(a_{N+1})\, p(a_{N+1}|\boldsymbol{t}_N)\, da_{N+1}$
- One can compute the convolution of a Gaussian and a sigmoid function:

$$\int \sigma(a)\mathcal{N}(a|\mu,\sigma^2)da \simeq \sigma\left(\kappa(\sigma^2)\mu\right)$$

$$\kappa(\sigma^2)=\sqrt{1+\frac{\pi\sigma^2}{8}}$$

We can thus try to approximate the posterior distribution $p(a_{N+1}|\boldsymbol{t}_N)$ as Gaussian

- How do we approximate the posterior?
  - Variational inference and make use of the local variational bound on the logistic sigmoid.
  - Expectation propagation
  - Laplace approximation

Laplace Approximation

- We can rewrite the posterior over a_{N+1} using Bayes' theorem:

$$
\begin{aligned}
p(a_{N+1} \mid \boldsymbol{t}_N) &= \int p(a_{N+1}, \boldsymbol{a}_N \mid \boldsymbol{t}_N)\, d\boldsymbol{a}_N \\
&= \frac{1}{p(\boldsymbol{t}_N)} \int p(a_{N+1}, \boldsymbol{a}_N)\, p(\boldsymbol{t}_N \mid a_{N+1}, \boldsymbol{a}_N)\, d\boldsymbol{a}_N \\
&= \frac{1}{p(\boldsymbol{t}_N)} \int p(a_{N+1} \mid \boldsymbol{a}_N)\, p(\boldsymbol{a}_N)\, p(\boldsymbol{t}_N \mid \boldsymbol{a}_N)\, d\boldsymbol{a}_N \\
&= \int p(a_{N+1} \mid \boldsymbol{a}_N)\, p(\boldsymbol{a}_N \mid \boldsymbol{t}_N)\, d\boldsymbol{a}_N
\end{aligned}
$$

- We know how to compute mean and covariance for $p(a_N|\boldsymbol{a}_N)$.

$$
p(a_{N+1} \mid \boldsymbol{a}_N) = \mathcal{N}\left(\boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{a}_N,\, c - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k}\right)
$$

- It remains to find a Gaussian approximation only for $p(\boldsymbol{a}_N|\boldsymbol{t}_N)$

- This is done noting that $p(\boldsymbol{a}_N \mid \boldsymbol{t}_N) \propto p(\boldsymbol{t}_N \mid \boldsymbol{a}_N)\, p(\boldsymbol{a}_N)$

Laplace Approximation

- We rewrote the posterior over a_{N+1} using Bayes' theorem:

$$p(a_{N+1} \mid \boldsymbol{t}_N) = \int p(a_{N+1} \mid \boldsymbol{a}_N)\, p(\boldsymbol{a}_N \mid \boldsymbol{t}_N)\, d\boldsymbol{a}_N$$

where

$$p(a_{N+1} \mid \boldsymbol{a}_N) = \mathcal{N}\left(\boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{a}_N, c - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k}\right)$$

- Also using $p(\boldsymbol{a}_N \mid \boldsymbol{t}_N) \propto p(\boldsymbol{t}_N \mid \boldsymbol{a}_N) p(\boldsymbol{a}_N)$, $p(\boldsymbol{a}_N) = \mathcal{N}\left(0, \boldsymbol{C}_N\right)$ and

$$p(\boldsymbol{t}_N \mid \boldsymbol{a}_N) = \prod_{n=1}^{N} \sigma(a_n)^{t_n} \left(1 - \sigma(a_n)\right)^{1-t_n} = \prod_{n=1}^{N} e^{a_n t_n} \sigma(-a_n)$$

where in the last derivation we used: $\sigma(a_n) = \dfrac{1}{1+e^{-a_n}}$ and

$$\sigma(a_n)^{t_n} \left(1 - \sigma(a_n)\right)^{1-t_n} = \underbrace{\left(1 - \sigma(a_n)\right)}_{\sigma(-a_n)} \left(\underbrace{\frac{\sigma(a_n)}{1 - \sigma(a_n)}}_{e^{a_n}} \right)^{t_n} = e^{a_n t_n} \sigma(-a_n)$$

Laplace Approximation

- The log of $p(\boldsymbol{a}_N \mid \boldsymbol{t}_N) \propto p(\boldsymbol{t}_N \mid \boldsymbol{a}_N) p(\boldsymbol{a}_N)$ which up to an additive constant is:

$$p(\boldsymbol{a}_N \mid \boldsymbol{t}_N) \propto p(\boldsymbol{a}_N) p(\boldsymbol{t}_N \mid \boldsymbol{a}_N)$$

$$\propto \mathcal{N}(0, \boldsymbol{C}_N) \prod_{n=1}^{N} e^{a_n t_n} \underbrace{\sigma(-a_n)}_{1/(1+e^{a_n})} \Rightarrow$$

$$\Psi(\boldsymbol{a}_N) = \ln p(\boldsymbol{a}_N \mid \boldsymbol{t}_N) = \ln p(\boldsymbol{t}_N \mid \boldsymbol{a}_N) + \ln p(\boldsymbol{a}_N)$$

$$= -\frac{1}{2}\boldsymbol{a}_N^T \boldsymbol{C}_N^{-1} \boldsymbol{a}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T \boldsymbol{a}_N - \sum_{n=1}^{N} \ln\left(1 + e^{a_n}\right)$$

- We will need to compute the 2nd derivative of this to come up with the Laplace approximation.

Laplace Approximation

$$\Psi(\boldsymbol{a}_N) = -\frac{1}{2}\boldsymbol{a}_N^T\boldsymbol{C}_N^{-1}\boldsymbol{a}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T\boldsymbol{a}_N - \sum_{n=1}^{N}\ln\left(1+e^{a_n}\right)$$

- The gradients are given as

$$\nabla\Psi(\boldsymbol{a}_N) = \boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N, \quad \boldsymbol{\sigma}_N = \left(\sigma(a_1),...,\sigma(a_N)\right)^T$$

$$\nabla^2\Psi(\boldsymbol{a}_N) = -\boldsymbol{W}_N - \boldsymbol{C}_N^{-1}, \quad \boldsymbol{W}_N = diag\left(\sigma(a_1)(1-\sigma(a_1)),...,\sigma(a_N)(1-\sigma(a_N))\right)$$

- We cannot find the mode by setting $\nabla\Psi(\boldsymbol{a}_N) = 0$ since $\boldsymbol{\sigma}_N$ depends nonlinearly on $\boldsymbol{a}_N$. For this we will use the IRLS algorithm.
- Note that $0 < \sigma(a_1)(1-\sigma(a_1)) < 1/4 \Rightarrow \boldsymbol{W}_N$ is positive definite.
- The Hessian $A = -\nabla^2\Psi(\boldsymbol{a}_N)$ is positive definite and thus the posterior $p(\boldsymbol{a}_N \mid \boldsymbol{t}_N)$ is log convex and has a single global mode.
- At the mode, the gradient vanishes giving: $\boldsymbol{a}_N^* = \boldsymbol{C}_N(\boldsymbol{t}_N - \boldsymbol{\sigma}_N)$

Iterative Reweighted Least Squares

- We can use the iterative reweighted least squares (IRLS) algorithm ($\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \boldsymbol{H}^{-1}\nabla E(\boldsymbol{w})$) to update for $\boldsymbol{a}_N$ (find the minimum of $-\ln p(\boldsymbol{a}_N \mid \boldsymbol{t}_N)$).

- The iterative update equation for $\boldsymbol{a}_N$ is:

$$\boldsymbol{a}_N^{new} = \boldsymbol{C}_N(\boldsymbol{I} + \boldsymbol{W}_N\boldsymbol{C}_N)^{-1}\left(\boldsymbol{t}_n - \boldsymbol{\sigma}_n + \boldsymbol{W}_N\boldsymbol{a}_N\right)$$

Here we used

$$\begin{aligned}
\boldsymbol{a}_N^{new} &= \boldsymbol{a}_N + (\boldsymbol{W}_N + \boldsymbol{C}_N^{-1})^{-1}\left(\boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N\right) = \boldsymbol{a}_N + \boldsymbol{C}_N(\boldsymbol{W}_N\boldsymbol{C}_N + \boldsymbol{I})^{-1}\left(\boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N\right) \\
&= \boldsymbol{C}_N(\boldsymbol{W}_N\boldsymbol{C}_N + \boldsymbol{I})^{-1}\left\{(\boldsymbol{W}_N\boldsymbol{C}_N + \boldsymbol{I})\boldsymbol{C}_N^{-1}\boldsymbol{a}_N + \left(\boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N\right)\right\} \\
&= \boldsymbol{C}_N(\boldsymbol{W}_N\boldsymbol{C}_N + \boldsymbol{I})^{-1}\left\{\boldsymbol{W}_N\boldsymbol{a}_N + \boldsymbol{t}_N - \boldsymbol{\sigma}_N\right\}
\end{aligned}$$

Laplace Approximation

- We can use the iterative reweighted least squares (IRLS) algorithm ($\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \boldsymbol{H}^{-1}\nabla E(\boldsymbol{w})$) to update for $\boldsymbol{a}_N$.

- The iterative update equation for $\boldsymbol{a}_N$ is:

$$\boldsymbol{a}_N^{new} = \boldsymbol{C}_N(\boldsymbol{I} + \boldsymbol{W}_N\boldsymbol{C}_N)^{-1}\left(\boldsymbol{t}_n - \boldsymbol{\sigma}_n + \boldsymbol{W}_N\boldsymbol{a}_N\right)$$

- The mode position $\boldsymbol{a}_N^*$ and the Hessian matrix $\boldsymbol{H}$ at this position $\boldsymbol{H} = -\nabla^2\Psi(\boldsymbol{a}_N) = \boldsymbol{W}_N + \boldsymbol{C}_N^{-1}$ define the Gaussian approximation to $p(\boldsymbol{a}_N \mid \boldsymbol{t}_N)$

$$q(\boldsymbol{a}_N) = \mathcal{N}(\boldsymbol{a}_N \mid \boldsymbol{a}_N^*, \boldsymbol{H}^{-1})$$

In the Hessian note that $\boldsymbol{W}_N$ is computed using $\boldsymbol{a}_N^*$.

Laplace Approximation

- Now we can go back to the formulas for linear Gaussian models and compute the integrals

$$p(a_{N+1}\,|\,\boldsymbol{t}_N)=\int p(a_{N+1}\,|\,\boldsymbol{a}_N)\,p(\boldsymbol{a}_N\,|\,\boldsymbol{t}_N)\,d\boldsymbol{a}_N$$

$$p(a_{N+1}\,|\,\boldsymbol{a}_N)=\mathcal{N}\left(\boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{a}_N,\,c-\boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{k}\right)$$

$$p(\boldsymbol{a}_N\,|\,\boldsymbol{t}_N)\approx q(\boldsymbol{a}_N)=\mathcal{N}(\boldsymbol{a}_N\,|\,\boldsymbol{a}_N^*,\boldsymbol{H}^{-1}),\;\boldsymbol{H}=\boldsymbol{W}_N+\boldsymbol{C}_N^{-1}$$

from which we have:

$$p(a_{N+1}\,|\,\boldsymbol{t}_N)=\mathcal{N}\left(\boldsymbol{k}^T(\boldsymbol{t}_N-\boldsymbol{\sigma}_N),\,c-\boldsymbol{k}^T\left(\boldsymbol{W}_N^{-1}+\boldsymbol{C}_N\right)^{-1}\boldsymbol{k}\right)$$

- Finally, we can also compute $p(t_{N+1}|\boldsymbol{t})$ (or the decision boundary $p(t_{N+1}|\boldsymbol{t})=0.5$ – in this case $\mu=\boldsymbol{k}^T(\boldsymbol{t}_N-\boldsymbol{\sigma}_N)=0$)

$$p(t_{N+1}=1\,|\,\boldsymbol{t}_N)=\int\sigma(a_{N+1})\,p(a_{N+1}\,|\,\boldsymbol{t}_N)\,da_{N+1}$$

using $$\int\sigma(a)\mathcal{N}(a\,|\,\mu,\sigma^2)da\approx\sigma\left(\kappa(\sigma^2)\mu\right),\quad\kappa(\sigma^2)=\sqrt{1+\frac{\pi\sigma^2}{8}}$$

Learning the Hyperparameters

- To determine the parameters $\boldsymbol{\theta}$ of the covariance function, we can maximize the likelihood function:

$$p(\boldsymbol{t}_N \mid \boldsymbol{\theta}) = \int p(\boldsymbol{t}_N \mid \boldsymbol{a}_N)\, p(\boldsymbol{a}_N \mid \boldsymbol{\theta}) d\boldsymbol{a}_N = \int p(\boldsymbol{t}_N, \boldsymbol{a}_N \mid \boldsymbol{\theta}) d\boldsymbol{a}_N$$

- The integral is analytically intractable. The Laplace approximation can be applied again: $\ln p(\boldsymbol{t}_N, \boldsymbol{a}_N) = \Psi(\boldsymbol{a}_N) = -\frac{1}{2}\boldsymbol{a}_N^T \boldsymbol{C}_N^{-1} \boldsymbol{a}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T \boldsymbol{a}_N - \sum_{n=1}^{N} \ln\left(1 + e^{a_n}\right)$

$$\ln p(\boldsymbol{t}_N, \boldsymbol{a}_N) \approx \Psi(\boldsymbol{a}_N^*) - \frac{1}{2}(\boldsymbol{a}_N - \boldsymbol{a}_N^*)^T (W_N + C_N^{-1})(\boldsymbol{a}_N - \boldsymbol{a}_N^*)$$

- Integrating in $\boldsymbol{a}_N$ and using the normalization factor of a Gaussian gives:

$$\ln p(\boldsymbol{t}_N \mid \boldsymbol{\theta}) = \boldsymbol{\Psi}(\boldsymbol{a}_N^*) - \frac{1}{2}\ln|\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}| + \frac{N}{2}\ln(2\pi)$$

where

$$\boldsymbol{\Psi}(\boldsymbol{a}_N^*) = \ln p(\boldsymbol{a}_N^* \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{t}_N \mid \boldsymbol{a}_N^*)$$

- We now need an expression for the gradient of the log of $p(\boldsymbol{t}_N|\boldsymbol{\theta})$. Both $\boldsymbol{a}_N^*$ and $\boldsymbol{C}_N$ depend on θ

Learning the Hyperparameters

- To determine the parameters , maximize the likelihood:

$$\Psi(\boldsymbol{a}_N^*) = -\frac{1}{2}\boldsymbol{a}_N^{*T}\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T\boldsymbol{a}_N^* - \sum_{n=1}^{N}\ln\left(1+e^{a_n^*}\right)$$

$$\ln p(\boldsymbol{t}_N \mid \boldsymbol{\theta}) = \Psi\left(\boldsymbol{a}_N^*\right) - \frac{1}{2}\ln|\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}| + \frac{N}{2}\ln(2\pi)$$

- Using $\frac{\partial}{\partial x}\ln|\boldsymbol{A}| = Tr\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right)$ and $\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}$, we obtain

the following terms of $\frac{\partial}{\partial \theta_j}\ln p(\boldsymbol{t}_N \mid \theta)$ with explicit dependence on θ:

$$\frac{1}{2}\boldsymbol{a}_N^{*T}\boldsymbol{C}_N^{-1}\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{1}{2}Tr\left[\boldsymbol{C}_N^{-1}\left[\boldsymbol{I} - \left(\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}\right)^{-1}\boldsymbol{C}_N^{-1}\right]\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right] =$$

$$\frac{1}{2}\boldsymbol{a}_N^{*T}\boldsymbol{C}_N^{-1}\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{1}{2}Tr\left[\boldsymbol{C}_N^{-1}\left[\left(\boldsymbol{C}_N\boldsymbol{W}_N + \boldsymbol{I}\right)\left(\boldsymbol{C}_N\boldsymbol{W}_N + \boldsymbol{I}\right)^{-1} - \left(\boldsymbol{C}_N\boldsymbol{W}_N + \boldsymbol{I}\right)^{-1}\right]\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right]$$

$$\frac{1}{2}\boldsymbol{a}_N^{*T}\boldsymbol{C}_N^{-1}\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{1}{2}Tr\left[\left(I + \boldsymbol{C}_N\boldsymbol{W}_N\right)^{-1}\boldsymbol{W}_N\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right]$$

Learning the Hyperparameters

- To determine the parameters, maximize the likelihood:

$$\Psi(\boldsymbol{a}_N^*) = -\frac{1}{2}\boldsymbol{a}_N^{*T}\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T\boldsymbol{a}_N^* - \sum_{n=1}^{N}\ln\left(1+e^{a_n^*}\right)$$

$$\ln p(\boldsymbol{t}_N \mid \boldsymbol{\theta}) = \Psi\left(\boldsymbol{a}_N^*\right) - \frac{1}{2}\ln\left|\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}\right| + \frac{N}{2}\ln(2\pi)$$

- Using $\nabla\Psi(\boldsymbol{a}_N^*) = 0$, the term in $\frac{\partial}{\partial\theta_j}\ln p(\boldsymbol{t}_N \mid \theta)$ with explicit dependence on $\boldsymbol{a}_n^*$ is as follows:

$$-\frac{1}{2}\sum_{n=1}^{N}\frac{\partial \ln\left|\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}\right|}{\partial a_n^*}\frac{\partial a_n^*}{\partial\theta_j}$$

- Using $\frac{\partial}{\partial x}\ln|\boldsymbol{A}| = Tr\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right)$ and $\boldsymbol{W}_N = diag\left(\sigma(a_i)(1-\sigma(a_i))\right)$, $d\sigma/da_i = \sigma(a_i)(1-\sigma(a_i))$, it becomes

$$-\frac{1}{2}\sum_{n=1}^{N}Tr\left[\left(\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}\right)^{-1}\underbrace{\frac{\partial \boldsymbol{W}_N}{\partial\sigma(a_n)}}_{(1-2\sigma_n^*)}\underbrace{\frac{\partial\sigma(a_n)}{\partial a_n}}_{\sigma_n^*(1-\sigma_n^*)}\right]\frac{\partial a_n^*}{\partial\theta_j} = -\frac{1}{2}\sum_{n=1}^{N}\left[\left(I + \boldsymbol{C}_N\boldsymbol{W}_N\right)^{-1}\boldsymbol{C}_N\right]_{nn}\underset{\sigma(a_n^*)}{\sigma_n^*}(1-\sigma_n^*)(1-2\sigma_n^*)\frac{\partial a_n^*}{\partial\theta_j}$$

Learning Hyperparameters

- We finally can compute $\partial a_n^* / \partial \theta_j$ as follows : $\boldsymbol{a}_N^* = \boldsymbol{C}_N \left(\boldsymbol{t}_N - \boldsymbol{\sigma}_N \right) \Rightarrow$

$$\frac{\partial \boldsymbol{a}_N^*}{\partial \theta_j} = \frac{\partial \boldsymbol{C}_N}{\partial \theta_j} (\boldsymbol{t}_N - \boldsymbol{\sigma}_N) - \boldsymbol{C}_N \boldsymbol{W}_N \frac{\partial \boldsymbol{a}_N^*}{\partial \theta_j}$$

$$\boldsymbol{W}_N = diag\left(\sigma(a_i)\left(1-\sigma(a_i)\right) \right)$$

$$d\sigma / da_i = \sigma(a_i)(1-\sigma(a_i))$$

- From this we derive:

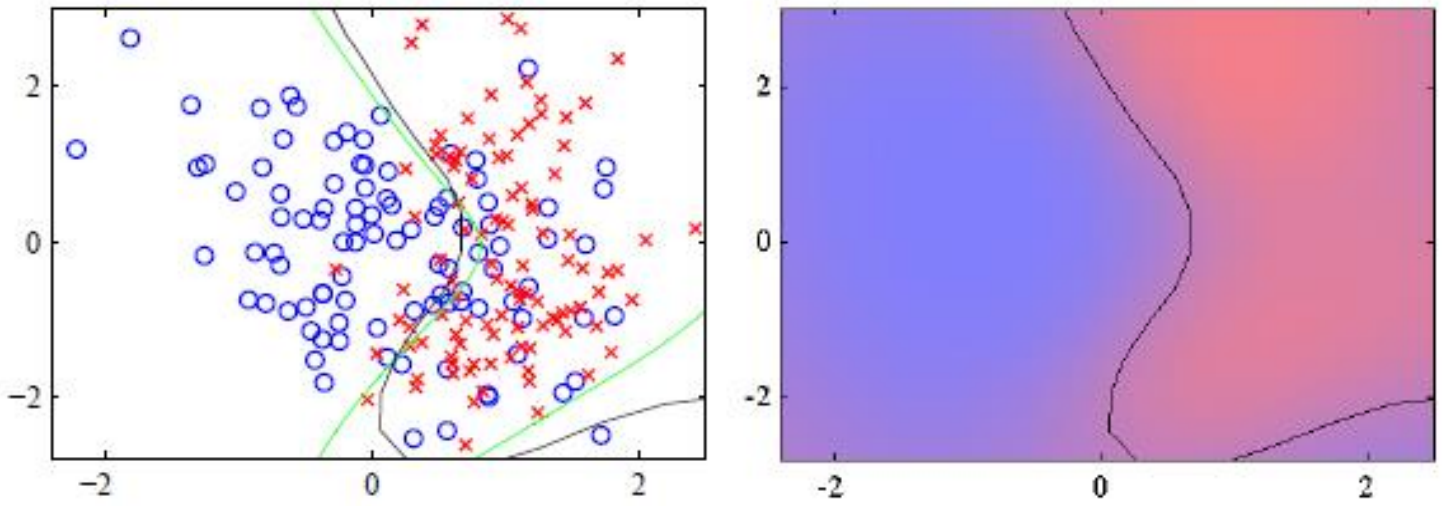
$$\frac{\partial \boldsymbol{a}_N^*}{\partial \theta_j} = \left(\boldsymbol{I} + \boldsymbol{C}_N \boldsymbol{W}_N \right)^{-1} \frac{\partial \boldsymbol{C}_N}{\partial \theta_j} (\boldsymbol{t}_N - \boldsymbol{\sigma}_N)$$

- We now have all the terms in the gradient $\frac{\partial}{\partial \theta_j} \ln p(\boldsymbol{t}_N \mid \theta)$ and can employ gradient optimization techniques.

Binary Classifier Based on Gaussian Process

- We now we have a binary classifier based on a Gaussian process. On left the data together with the optimal decision boundary from the true distribution in green, and the decision boundary from the Gaussian process classifier in black.

- On right the predicted posterior probability for the blue and red classes together with the Gaussian process decision boundary.



Requires installing NetLab and setting the appropriate path

MatLab Code

- Williams, C. K. I. and D. Barber (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 1342–1351.

Connections to Neural Networks

- Neural Networks
  - The range of representable functions is governed by the number M of hidden units
  - Within the maximum likelihood framework, they overfit as M comes close to the number of training samples
- Bayesian Neural Networks
  - The prior over $\boldsymbol{w}$ in conjunction with the network function $f(\boldsymbol{x}, \boldsymbol{w})$ produces a prior distribution over functions from $y(\boldsymbol{x})$
  - The distribution of functions will tend to a GP as $M \rightarrow \infty$.
    - In this limit the output variables of the neural network become independent.
    - However, the property that the outputs share hidden units and thus all of them affecting the values of the weights is lost in this limit.

- Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer. Lecture Notes in Statistics 118.

Connections to Neural Networks

- There are explicit forms for the covariance in the case of probit and Gaussian hidden unit activation functions.
  - These kernel functions $k(\boldsymbol{x}, \boldsymbol{x}')$ are nonstationary, i.e. not a function of $\boldsymbol{x} - \boldsymbol{x}'$.
  - This is a consequence of the Gaussian weight prior being centered on zero which breaks translation invariance in weight space.
- By working directly with the covariance function we have implicitly marginalized over the distribution of weights.
- If the weight prior is governed by hyperparameters, then their values will determine the length scales of the distribution over functions.

- Williams, C. K. I. (1998). Computation with infinite neural networks. *Neural Computation* **10**(5), 1203–1216.

Course Summary

Markov and Chebyshev Inequalities

- You can show (Markov's inequality) that if X is a non-negative integrable random variable and for any *a>0*:

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

$$Indeed: \quad \mathbb{E}[X] \geq \int_a^\infty x\pi(x)dx \geq a\int_a^\infty \pi(x)dx = a\Pr[X \geq a]$$

- You can generalize this using any function of the random variable *X* as:

$$\Pr[f(X) \geq a] \leq \frac{\mathbb{E}[f(X)]}{a}$$

- Using $f(X)=(X-\mathbb{E}[X])^2$, we derive the following Chebyshev inequality:

$$a \equiv \varepsilon^2, \mathbb{E}\left[\left(X-\mathbb{E}[X]\right)^2\right] = \sigma^2 \qquad \Pr\left[\left|X-\mathbb{E}[X]\right| \geq \varepsilon\right] \leq \frac{\sigma^2}{\varepsilon^2} \forall \varepsilon$$

- In terms of std's, we can restate as : $\Pr\left[\left|X-\mathbb{E}[X]\right| \geq \varepsilon\sigma\right] \leq \frac{1}{\varepsilon^2} \forall \varepsilon$

- Thus the probability of X being more than 2σ away from $\mathbb{E}[X]$ is $\leq 1/4$.

The Law of Large Numbers (LLN)

- Let X_i for $i = 1, 2, \ldots, n$ be independent and identically distributed random variables (i.i.d.) with $\mathbb{E}(X_i) = \mu$ and variance $Var(X_i) = \sigma^2$.

- Let $\overline{X_n} = \sum_{i=1}^{n} \frac{X_i}{n}$

- Note that $\mathbb{E}[\overline{X_n}] = \sum_{i=1}^{n} \frac{\mu}{n} = \mu$

$$Var[\overline{X_n}] = \frac{1}{n^2} \sum_{i=1}^{n} \sigma^2 = \frac{\sigma^2}{n}$$

- Weak LLN: $\lim_{n\to\infty} \Pr[|\overline{X_n} - \mu| \geq \varepsilon] = 0 \; \forall \varepsilon > 0$

- Strong LLN: $\lim_{n\to\infty} \overline{X_n} = \mu \; almost\; surely$

This means that with probability one, the average of any realizations of $x_1, x_2, \ldots$ of the random variables $X_1, X_2, \ldots$ converges to the mean.

MLE for a Multivariate Gaussian

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n \equiv \bar{x},\ \Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})(x_n - \mu_{ML})^T = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T - \bar{x}\bar{x}^T$$

- Note that *the unconstrained maximization of the log-likelihood gives a symmetric Σ.*

- As for the univariate case, we can define an unbiased covariance as:

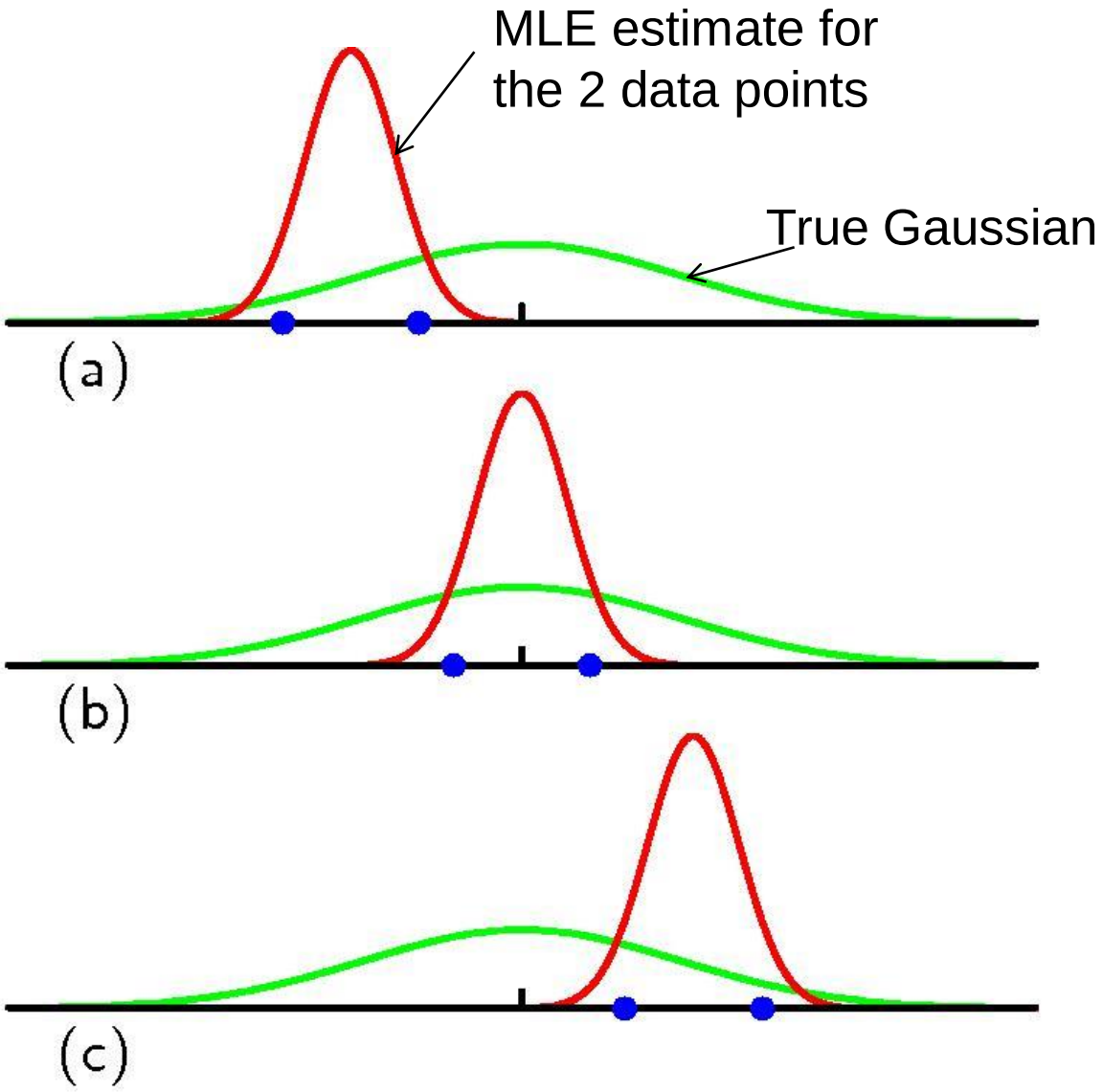
$$\bar{\Sigma}_{ML} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{ML})(x_n - \mu_{ML})^T,\ \mathbb{E}[\bar{\Sigma}_{ML}] = \Sigma$$

- To prove this, you will need to use that:

$$\mathbb{E}\left[\boldsymbol{x}_n \boldsymbol{x}_m^T\right] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \delta_{mn}\boldsymbol{\Sigma}$$

Bias in MLE

- In the schematic from Bishop's PRML, we consider 3 cases each with 2 data points extracted from the true Gaussian.
- The mean of the three distributions predicted via MLE (i.e. averaged over the data) is correct.
- However, the variance is underestimated since it is a variance with respect to the sample mean and NOT the true mean.

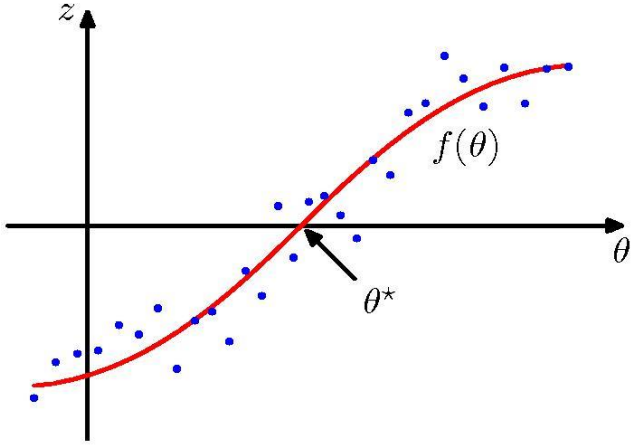


Robbins-Monro Algorithm

$$f(\theta) = \mathbb{E}(z \mid \theta) = \int z p(z \mid \theta) dz$$

- We want to find the root $f(\theta^*) = 0$ in a sequential manner: The Robbins-Monro algorithm proceeds as:

$$\theta^{(N)} = \theta^{(N-1)} - a_{N-1} z\left(\theta^{(N-1)}\right)$$



- The *learning coefficients* $\{a_N\}$ should satisfy:

$$\lim_{N\to\infty} a_N = 0, \sum_{n=1}^{\infty} a_N = \infty, \sum_{n=1}^{\infty} a_N^2 < \infty$$

Volume of a Sphere in High Dimensions

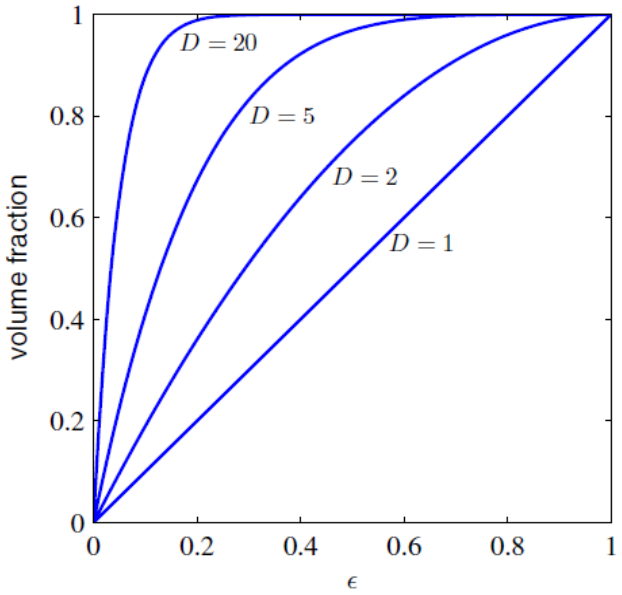
- Consider a sphere of radius $r = 1$ in D dimensions. Let us compute the fraction of the volume of the sphere that lies between radius $r = 1-\varepsilon$ and $r = 1$.

$$\frac{V_D(1)-V_D(1-\varepsilon)}{V_D(1)} = \frac{K_D 1^D - K_D(1-\varepsilon)^D}{K_D 1^D}$$

$$= 1-(1-\varepsilon)^D$$

- Note that for large D, this fraction tends to 1 even for small ε

- In high dimensions, the volume of the sphere is concentrated near the surface!



The volume of a sphere in D-dimensions is given as

$V_D(r) = K_D r^D$,

$K_D = D$ - dependent constant

Bayes' Theorem and Gaussian Linear Models

- Consider a linear Gaussian model: A Gaussian marginal distribution $p(\mathbf{x})$ and a Gaussian conditional distribution $p(\mathbf{y}|\mathbf{x})$ in which $p(\mathbf{y}|\mathbf{x})$ has a mean that is a linear function of $\mathbf{x}$, and a covariance which is independent of $\mathbf{x}$.

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}, \Lambda^{-1}\right)$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}\left(\mathbf{y} \mid \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}\right)$$

- *We want using Bayes' rule to find $p(\mathbf{y})$ and $p(\mathbf{x}|\mathbf{y})$.*

- We start with the joint distribution over $\mathbf{z}=(\mathbf{x},\mathbf{y})$ which is quadratic in the components of $\mathbf{z}$ – *so* $p(\mathbf{z})$ is a Gaussian.

Posterior Inference: Point Estimates

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{m(x)}$$

Maximum A Posteriori estimate (MAP)

$$\theta^* = \arg\max_{\theta} \log\left(\pi(\theta \mid x)\right) = \arg\max_{\theta} \left(\log \pi(x \mid \theta) + \log \pi(\theta)\right)$$

Posterior Mean

$$\hat{\theta} = \mathbb{E}_{p(\theta|x)}[\theta] = \int \theta \pi(\theta|x) d\theta$$

Posterior Quantiles

$$\Pr\left[\theta > a\right] = \int_a^{\infty} \pi(\theta \mid x) d\theta$$

Prediction

Suppose we have observed $\boldsymbol{x}$ and we want to make a prediction about (future) unknown observables: What is the probability of observing data $\hat{\boldsymbol{x}}$ If we already have observed data $\boldsymbol{x}$?

This means finding $g(\hat{\boldsymbol{x}}|\boldsymbol{x})$

We have:

$$g(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \int g(\hat{\boldsymbol{x}}, \theta|\boldsymbol{x})d\theta = \int \frac{\pi(\hat{\boldsymbol{x}}, \theta, \boldsymbol{x})}{m(\boldsymbol{x})} d\theta = \int \frac{\pi(\hat{\boldsymbol{x}}, \theta, \boldsymbol{x})}{\phi(\theta, \boldsymbol{x})} \frac{\phi(\theta, \boldsymbol{x})}{m(\boldsymbol{x})} d\theta =$$

$$= \int f(\hat{\boldsymbol{x}}|\theta, \boldsymbol{x})\pi(\theta|\boldsymbol{x})d\theta = \int f(\hat{\boldsymbol{x}}|\theta)\pi(\theta|\boldsymbol{x})d\theta$$

Compare this with the normalizing factor:

$$m(\hat{\boldsymbol{x}}) = \int f(\hat{\boldsymbol{x}}|\theta)\pi(\theta)d\theta$$

Marginal Likelihood or Evidence

- In the Bayesian parametric model, we define the following:
  - The joint distribution of (θ, X)

$$\phi(\theta, x) = \pi(\theta) f(x \mid \theta)$$

  - The marginal distribution of X

$$m(x) = \int \phi(\theta, x) d\theta = \int \pi(\theta) f(x \mid \theta) d\theta$$

  - For a realization $X=x$, $m(x)$ is called marginal likelihood or evidence

Normalizing Factor in Bayes' Rule

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{m(x)}$$

- What about *m(x)* (the marginal distribution on the data *x*)

$$m(x) = \int \phi(x, \theta) d\theta = \int f(x \mid \theta)\pi(\theta) d\theta$$

- It is the normalizing *constant of the posterior* $\pi(\theta \mid x)$
- It is not needed for inference or prediction
- It is essential for model validation (we will discuss this later on)

Predictive Distribution and Approximations

- Given the prior $\pi(\theta)$ and the likelihood $\ell(\theta|x)=f(x|\theta)$, Bayes' formula yields:[a]

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{\int f(x \mid \theta)\pi(\theta)d\theta}$$

- This represents all the information on θ that can be extracted from *x*.

- Note the integral at the demoninator of the Bayes' rule.

- The predictive distribution of *Y* when *Y*~*g*(*y*|θ,*x*) is (we will revisit this in a forthcoming lecture)

$$g(y \mid x) = \int g(y, \theta \mid x)d\theta = \int \underbrace{g(y \mid \theta, x)}_{Likelihood} \underbrace{\pi(\theta \mid x)}_{Posterior} d\theta$$

- This is to distinguish from prediction using $\hat{\theta} = \theta^{MLE}$ or $\hat{\theta} = \theta^{MAP}$: $g(y|\hat{\theta}, x)$

[a] Probability, Conditional Probability and Bayes Formula, B. Vidakovic

Exponential Family

- The exponential family of distributions over $\boldsymbol{x}$, given parameters $\boldsymbol{\eta}$, is defined to be the set of distributions of the form

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} \; or$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}, \; where: A(\boldsymbol{\eta}) = -\log g(\boldsymbol{\eta})$$

$\boldsymbol{x}$ is scalar/vector, discrete/continuous. ***η are the natural parameters and u(x) is referred to as a sufficient statistic.***

- *g(η) ensures that the distribution is normalized* and satisfies

$$g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x} = 1$$

- The normalization factor Z and the log of it A are defined as:

$$Z(\boldsymbol{\eta}) = \frac{1}{g(\boldsymbol{\eta})}, \; A(\boldsymbol{\eta}) = \ln Z(\boldsymbol{\eta}) = -\ln g(\boldsymbol{\eta}) = \ln \int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x}$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}/Z(\boldsymbol{\eta})$$

- The space of η for which $\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x} < \infty$ is the ***natural parameter space***.

Conjugate Priors

- For any member of the exponential family,

$$p(\boldsymbol{x}\,|\,\boldsymbol{\theta}) = h(\boldsymbol{x})\,g(\boldsymbol{\eta}(\boldsymbol{\theta}))\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\,u(\boldsymbol{x})\right\}$$

there exists a conjugate prior that can be written in the form

$$p(\boldsymbol{\theta}|\nu_0,\boldsymbol{\tau}_0) \propto g(\boldsymbol{\eta}(\boldsymbol{\theta}))^{\nu_0}\exp\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\boldsymbol{\tau}_0\} = \exp\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_0 - A(\boldsymbol{\eta}(\boldsymbol{\theta}))\nu_0\},\ where:\ \boldsymbol{\tau}_0 \equiv \nu_0\bar{\boldsymbol{\tau}}_0$$

- In normalized form, we write:

$$p(\boldsymbol{\theta}|\nu_0,\boldsymbol{\tau}_0) = \frac{1}{Z(\nu_0,\boldsymbol{\tau}_0)}g(\boldsymbol{\eta}(\boldsymbol{\theta}))^{\nu_0}\exp\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\boldsymbol{\tau}_0\} = \frac{1}{Z(\nu_0,\boldsymbol{\tau}_0)}\exp\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\boldsymbol{\tau}}_0 - A(\boldsymbol{\eta}(\boldsymbol{\theta}))\nu_0\}$$

$$where: Z(\nu_0,\boldsymbol{\tau}_0) = \int \exp\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\boldsymbol{\tau}}_0 - A(\boldsymbol{\eta}(\boldsymbol{\theta}))\nu_0\}d\boldsymbol{\theta}$$

Standard Exponential Families

f(x|θ)	π(θ)	π(θ|x)
Normal $N(\theta, \sigma^2)$	Normal $\mathcal{N}(\mu, \tau^2)$	$\mathcal{N}(\rho(\sigma^2\mu + \tau^2 x), \rho\sigma^2\tau^2)$ $\rho^{-1} = \sigma^2 + \tau^2$
Poisson $\mathcal{P}(\theta)$	Gamma $\mathcal{P}(\theta)\mathcal{G}(\alpha, \beta)$	$\mathcal{G}(\alpha + x, \beta + 1)$
Gamma $\mathcal{G}(\nu, \theta)$	Gamma $\mathcal{G}(\alpha, \beta)$	$\mathcal{G}(\alpha + \nu, \beta + x)$
Binomial $\mathcal{B}(\mathrm{n}, \theta)$	Beta $\mathcal{B}\mathrm{e}(\alpha, \beta)$	$\mathcal{B}\mathrm{e}(\alpha + x, \beta + n - x)$
Negative Binomial $\mathcal{N}\mathrm{eg}(\mathrm{m}, \theta)$	Beta $\mathcal{B}\mathrm{e}(\alpha, \beta)$	$\mathcal{B}\mathrm{e}(\alpha + m, \beta + x)$
Multinomial $\mathcal{M}_k(\theta_1, \ldots, \theta_k)$	Dirichlet $\mathcal{D}(\alpha_1, \ldots, \alpha_k)$	$\mathcal{D}(\alpha_1 + x_1, \ldots, \alpha_k + x_k)$
Normal $\mathcal{N}(\mu, 1/\theta)$	Gamma $\mathcal{G}\mathrm{a}(\alpha, \beta)$	$\mathcal{G}(\alpha + 0.5, \beta + (\mu - x)^2/2)$

Generalizing to Multivariate Gaussians

- Suppose you have $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N \sim$(i.i.d) $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$
- We do not know $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$
- When both sets of parameters are unknown, a conjugate family of priors is one in which

$$\boldsymbol{\Sigma} \sim \mathcal{IW}(\nu, \boldsymbol{\Lambda}^{-1})$$

and

$$\boldsymbol{\mu} \mid \boldsymbol{\Sigma} \sim \mathcal{N}(\boldsymbol{\eta}, \boldsymbol{\Sigma} / \kappa)$$

- The Wishart distribution is a multivariate analog of the Gamma *distribution*. If matrix *U* has the Wishart distribution, then U^{-1} has the inverse-Wishart distribution.
- The quantity ν *is a positive scalar, while* $\boldsymbol{\Lambda}$ *is a* positive definite matrix. They play roles analogous to those played by α and β, respectively, in the Gamma *distribution*.
- The other parameters of the prior are the mean vector $\boldsymbol{\eta}$ and κ, the latter of which represents the ``a priori number of observations''.

Wishart Distribution

Wishart	$W \sim \text{Wishart}_\nu(S)$ $p(W) = \text{Wishart}_\nu(W\|S)$ (implicit dimension $k \times k$)	degrees of freedom ν symmetric, pos. definite $k \times k$ scale matrix S

$$p(W) = \left(2^{\nu k/2}\pi^{k(k-1)/4}\prod_{i=1}^{k}\Gamma\left(\frac{\nu+1-i}{2}\right)\right)^{-1} \times |S|^{-\nu/2}|W|^{(\nu-k-1)/2} \times \exp\left(-\frac{1}{2}\text{tr}(S^{-1}W)\right), \; W \text{ pos. definite}$$

$$\text{E}(W) = \nu S$$

Gamma	$\theta \sim \text{Gamma}(\alpha, \beta)$ $p(\theta) = \text{Gamma}(\theta\|\alpha, \beta)$	shape $\alpha > 0$ inverse scale $\beta > 0$

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}, \quad \theta > 0$$

$$\text{E}(\theta) = \frac{\alpha}{\beta}$$
$$\text{var}(\theta) = \frac{\alpha}{\beta^2}$$
$$\text{mode}(\theta) = \frac{\alpha-1}{\beta}, \text{ for } \alpha \geq 1$$

Bayesian Data Analysis, A. Gelman, J. Carlin, H. Stern and D. Rubin, 2004

Jeffrey's Noninformative Priors

- Jeffrey's proposes a more intrinsic approach which avoids the need to take the invariance structure into account.

- Given a likelihood $f(x \mid \theta)$, Jeffrey's noninformative prior distributions are based on Fisher information, given by

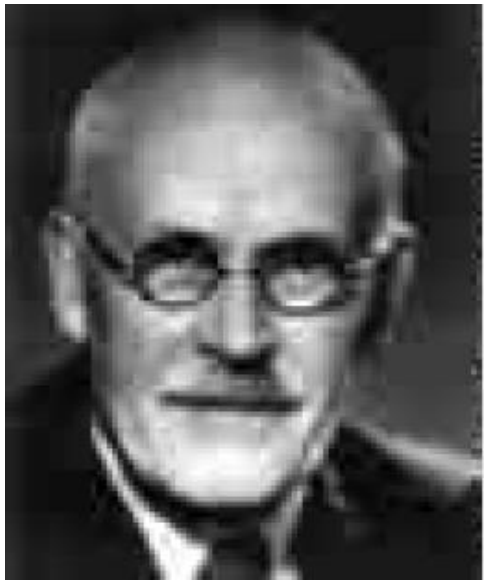
$$I(\theta) = \mathbb{E}_{X|\theta}\left(\frac{\partial \log f(X \mid \theta)}{\partial \theta} \frac{\partial \log f(X \mid \theta)^T}{\partial \theta} \right) = -\mathbb{E}_{X|\theta}\left(\frac{\partial^2 \log f(X \mid \theta)}{\partial \theta^2} \right)$$

the corresponding prior distribution is

$$\pi(\theta) \propto \left| I(\theta) \right|^{1/2}$$

Determinant of I

Sir Harold Jeffreys
(1891–1989)



Empirical Bayes – Evidence Approximation

- In hierarchical Bayesian models, we need to compute the posterior on multiple levels of latent variables. For example, in a two-level model, we need to compute

$$p(\boldsymbol{\eta}, \boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \boldsymbol{\eta})\, p(\boldsymbol{\eta})$$

- In some cases, we can analytically marginalize out $\boldsymbol{\theta}$; this leaves is with the simpler problem of just computing $p(\boldsymbol{\eta} \mid \mathcal{D})$.
- As a computational shortcut, we can *approximate the posterior on the hyper-parameters with a point-estimate,*

$$p(\boldsymbol{\eta}|\mathcal{D}) \approx \delta_{\overline{\boldsymbol{\eta}}}(\boldsymbol{\eta}),\ \overline{\boldsymbol{\eta}} = \operatorname{argmax} p(\boldsymbol{\eta}|\mathcal{D}) = \operatorname{argmax} \left[\int p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\eta}) d\boldsymbol{\theta} \right]$$

- Since $\boldsymbol{\eta}$ is typically much smaller than $\boldsymbol{\theta}$ in dimensionality, it is less prone to overfitting, so we can safely use a uniform prior on $\boldsymbol{\eta}$.

- The quantity inside the brackets is *the marginal or integrated likelihood, often called the evidence.* This overall approach is called *empirical Bayes (EB) or type-II maximum likelihood.* In machine learning, it is sometimes called the *evidence procedure*.

Empirical Bayes

- *Empirical Bayes violates the principle that the prior should be chosen independently of the data.*

- We can just view it as a cheap approximation to inference in a hierarchical Bayesian model, just as we viewed MAP estimation as an approximation to inference in the one level model $\boldsymbol{\theta} \rightarrow \mathcal{D}$.

- We can construct a hierarchy in which the more integrals one performs, the "more Bayesian" one becomes:

Method	Definition
Maximum likelihood	$\hat{\theta} = \text{argmax}_{\boldsymbol{\theta}}\, p(\mathcal{D}\|\theta)$
MAP estimation	$\hat{\theta} = \text{argmax}_{\boldsymbol{\theta}}\, p(\mathcal{D}\|\theta)p(\theta\|\eta)$
ML-II (Empirical Bayes)	$\hat{\eta} = \text{argmax}_{\boldsymbol{\eta}} \int p(\mathcal{D}\|\theta)p(\theta\|\eta)d\theta = \text{argmax}_{\boldsymbol{\eta}}\, p(\mathcal{D}\|\eta)$
MAP-II	$\hat{\eta} = \text{argmax}_{\boldsymbol{\eta}} \int p(\mathcal{D}\|\theta)p(\theta\|\eta)p(\eta)d\theta = \text{argmax}_{\boldsymbol{\eta}}\, p(\mathcal{D}\|\eta)p(\eta)$
Full Bayes	$p(\theta, \eta\|\mathcal{D}) \propto p(\mathcal{D}\|\theta)p(\theta\|\eta)p(\eta)$

Bayesian Model Validation

From data we can learn the parameters for each model and then the model itself

$$\boldsymbol{x} \Rightarrow \pi_i(\theta_i \mid \boldsymbol{x}, M_i) = \frac{f_i(\boldsymbol{x} \mid \theta_i, M_i)\pi_i(\theta_i \mid M_i)}{\pi_i(\boldsymbol{x} \mid M_i)} \Rightarrow \quad \pi_i(M_i \mid \boldsymbol{x}) = \frac{\pi_i(\boldsymbol{x} \mid M_i)\pi_i(M_i)}{\pi(\boldsymbol{x})}$$

Noting that

$$\pi_i(\boldsymbol{x} \mid M_i) = \int f_i(\boldsymbol{x} \mid \theta_i, M_i)\pi_i(\theta_i \mid M_i)d\theta_i$$

we can find the best model that represents the data by computing:

$$\frac{\pi(M_1 \mid \boldsymbol{x})}{\pi(M_2 \mid \boldsymbol{x})} = \frac{\pi(\boldsymbol{x} \mid M_1)\pi(M_1)}{\pi(\boldsymbol{x} \mid M_2)\pi(M_2)} = \underbrace{\frac{\int f_1(\boldsymbol{x} \mid \theta_1, M_1)\pi_1(\theta_1 \mid M_1)d\theta_1}{\int f_2(\boldsymbol{x} \mid \theta_2, M_2)\pi_2(\theta_2 \mid M_2)d\theta_2}}_{B_{10}^{\pi}} \underbrace{\frac{\pi(M_1)}{\pi(M_2)}}_{\text{Ratio of Priors}}$$

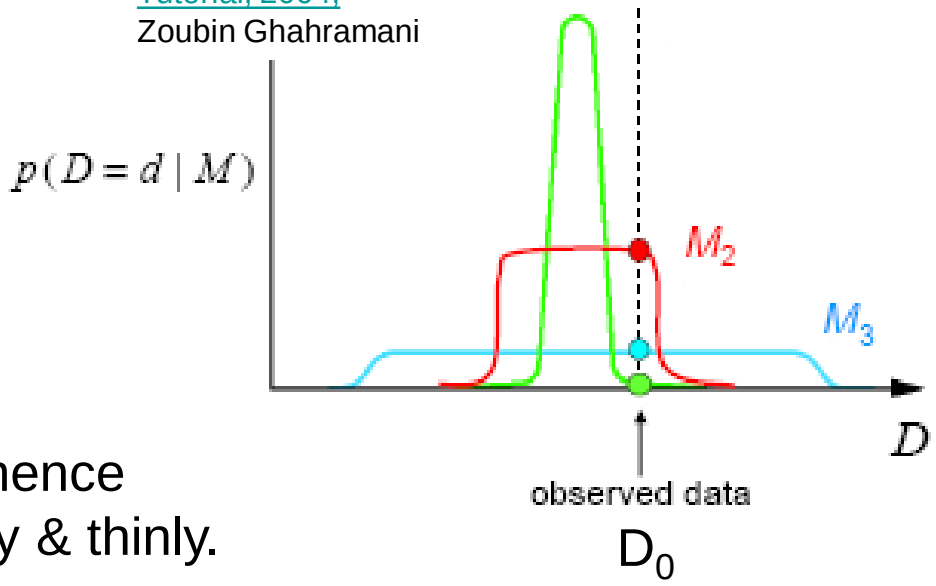
B_{10}^{π}: Ratio of Bayes' factors

Bayesian Occam's Razor

- To further understand the Bayesian Occam's razor effect is to note that probabilities must sum to one (sum over all possible data sets)

$$\sum_{\mathcal{D}'} p\left(\mathcal{D}' \mid m\right) = 1$$

Bayesian Methods for Machine Learning, ICML Tutorial, 2004, Zoubin Ghahramani



- Model 1 is too simple and assigns low probability to D_0.

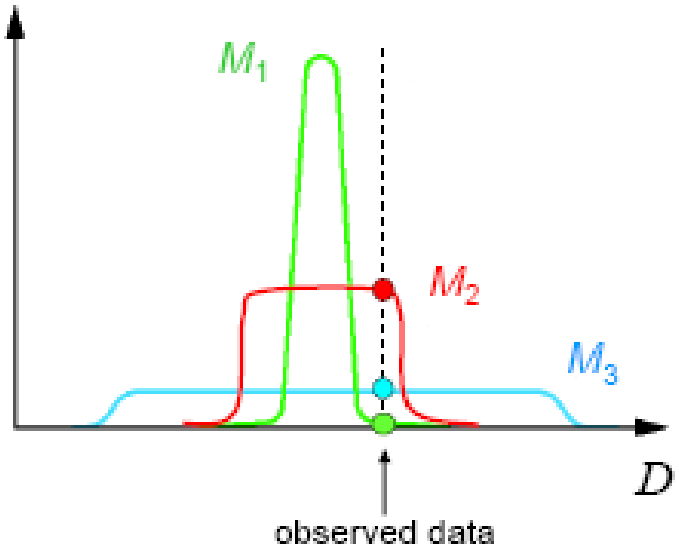
- Model 3 also assigns D_0 relatively low probability, because it can predict many data sets, and hence it spreads its probability quite widely & thinly.

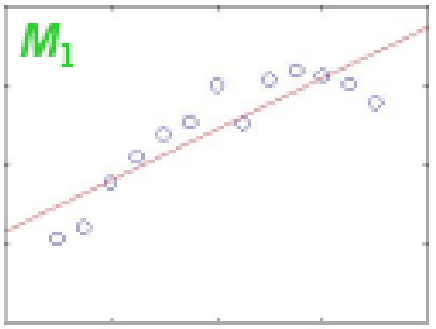
- Model 2 is "just right": it predicts the observed data with a reasonable degree of confidence, but does not predict too many other things. Hence model 2 is the most probable model.

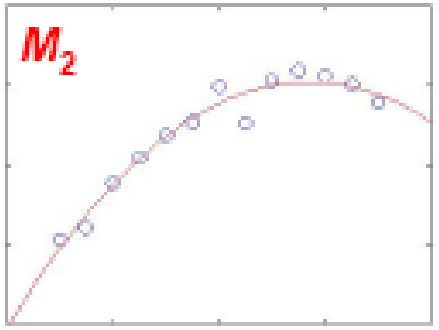
Bayesian Occam's Razor

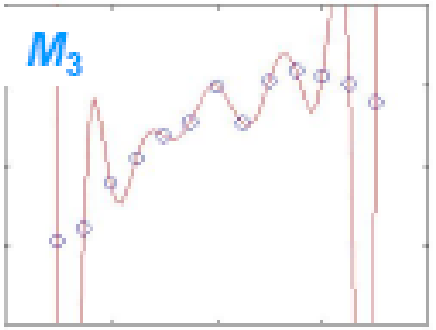
Bayesian Methods for Machine Learning, ICML Tutorial, 2004, Zoubin Ghahramani

$p(D = d \mid M)$









M_1 : the too simple model is unlikely to generate this data

M_3 : the too complex model explains poorly a lots of data sets and it is a little better but still unlikely to have generated our data

M_2 : the just right model has the highest marginal likelihood

Laplace Approximation

- The Laplace approximation allows a Gaussian approximation of the parameter posterior about the maximum a posteriori (MAP) parameter estimate.
- Consider a data set $\mathcal{D}$ and M models $\mathcal{M}_i$, $i=1,..,M$ with corresponding parameters $\boldsymbol{\theta}_i$, i=1,…M. We compare models using the posteriors:

$$p(\mathcal{M} \mid \mathcal{D}) \propto p(\mathcal{M}) p(\mathcal{D} \mid \mathcal{M})$$

- For large sets of data $\mathcal{D}$ (relative to the model parameters), the parameter posterior is approximately Gaussian around the MAP estimate $\boldsymbol{\theta}_m^{MAP}$ (can also use 2nd order Taylor expansion of the log-posterior):

$$p(\boldsymbol{\theta}_m \mid \mathcal{D}, \mathcal{M}_m) \approx (2\pi)^{-d/2} |\mathbf{A}|^{1/2} \exp\left(-\frac{1}{2} \left(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^{MAP}\right)^T \mathbf{A} \left(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^{MAP}\right) \right),$$

$$A_{ij} = -\left. \frac{\partial^2 \log P(\boldsymbol{\theta}_m \mid \mathcal{D}, \mathcal{M}_m)}{\partial \boldsymbol{\theta}_{mi} \partial \boldsymbol{\theta}_{mj}} \right|_{\boldsymbol{\theta}_m^{MAP}}$$

Sampling From an Arbitrary Distribution

- Consider an arbitrary probability density $\pi(\mathbf{x})$

- Monte Carlo approximation is given by

$$\hat{\pi}_N(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}}(\mathbf{x}) \text{ , } where\ X^{(i)} \overset{i.i.d.}{\sim} \pi$$

- For any function $f : \mathcal{X} \to \mathbb{R}$

$$\mathbb{E}_{\hat{\pi}_N}(f) = \frac{1}{N}\sum_{i=1}^{N} f(X^{(i)}) \cong \mathbb{E}_{\pi}(f)$$

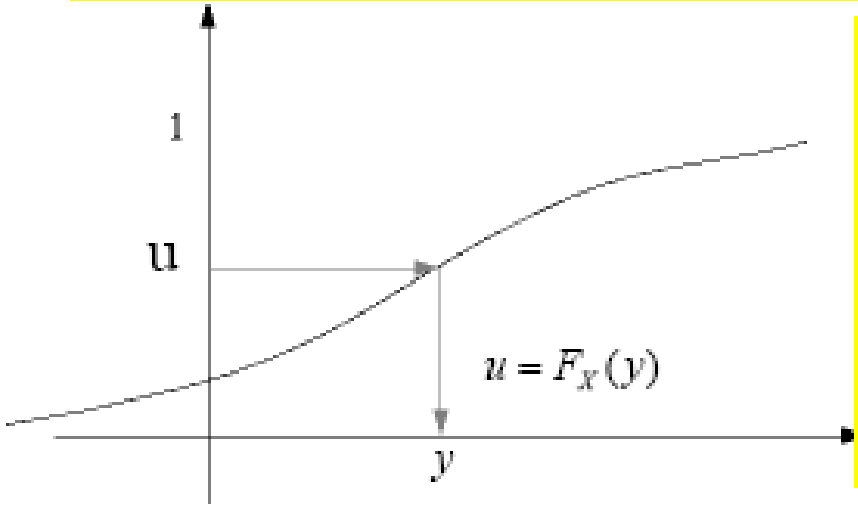
But how do we sample from an arbitrary distribution?

or more precisely:

$$\mathbb{E}_X\left[\mathbb{E}_{\hat{\pi}_N}(f)\right] = \mathbb{E}_{\pi}(f)\ and\ Var_X\left(\mathbb{E}_{\hat{\pi}_N}(f)\right) = \frac{Var_{\pi}\ (f)}{N}$$

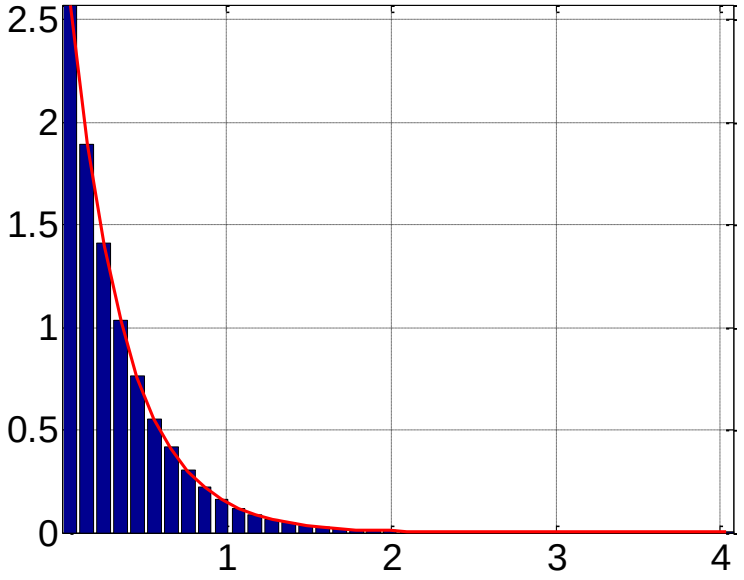
Inverse Method: Exponential Distribution

Exponential(λ): $\pi(x)= \lambda e^{-\lambda x}$, $F_X(y) = 1 - e^{-\lambda y}$, $x= -\ln(1-u)/\lambda$



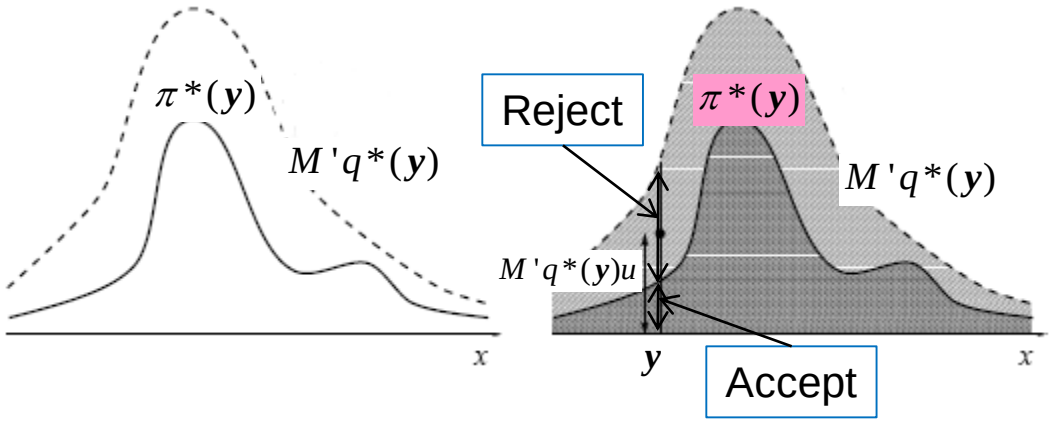
As *F(y)* is area under *π(y)*, *$y = F^{-1}(u)$* prescribes that

- **Choose *u=(0,1]*, then find value y that has that *fraction u of area to the left of y, or u=F(y)***
- **Return that value of *x=y*.**



MatLab Implementation

Rejection Sampling



- Set *i=1*
- Repeat until *i=N*
  - Sample $y \sim q(y)$ *and* $u \sim \mathcal{U}_{(0,1)}$
  - If $u < \frac{\pi^*(y)}{M'q^*(y)}$ then accept (set $x^{(i)}=y$) and increment the counter i
    - ✓ Otherwise, reject

The distribution $\pi(y)$ needs to be known only up to a normalizing constant:

$$\pi(y) = \frac{\pi^*(y)}{Z}, \; Z = \int \pi^*(y)dy$$

Importance Sampling

- Computing $\mathbb{E}_{\pi}\{f(X)\}$ is transformed to computing

$$\mathbb{E}_{q}\left\{\underbrace{\frac{f(X)\pi(X)}{q(X)}}_{f_q(X)}\right\}$$

- We can employ MC for this task:
  - Generate samples $X^{(i)} \sim q(x)\,(i.i.d.)$ and evaluate:

$$\mathbb{E}_{\hat{q}_N}(f_q(x)) = \frac{1}{N}\sum_{i=1}^{N} f_q(X^{(i)}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\pi(X^{(i)})}{q(X^{(i)})} f(X^{(i)}) = \frac{1}{N}\sum_{i=1}^{N} w(X^{(i)}) f(X^{(i)})$$

where the importance weight is $w(X^{(i)}) \equiv \frac{\pi(X^{(i)})}{q(X^{(i)})}$.

Asymptotic Variance of the Normalized IS

- It follows that

$$Var\left(\mathbb{E}_{\hat{\pi}_N}(f(X))\right) \simeq \sigma_A^2\left(\frac{\partial g}{\partial A}(\mu)\right)^2 + \sigma_B^2\left(\frac{\partial g}{\partial B}(\mu)\right)^2 + 2\frac{\partial g}{\partial A}(\mu)\frac{\partial g}{\partial B}(\mu)\sigma_{A,B} =$$

$$= \frac{\sigma_A^2}{\mu_B^2} + \frac{\sigma_B^2\mu_A^2}{\mu_B^4} - 2\frac{\sigma_{A,B}\mu_A}{\mu_B^3}$$

- Straight forward substitution of the expressions for $\sigma_A^2, \sigma_B^2, \mu_A, \mu_B$, and using $\mathbb{E}_q\left(\mathbb{E}_{\hat{\pi}_N}(f(X))\right) \simeq \mathbb{E}_\pi(f(X))$ leads to the following CLT asymptotic result:

$$\sqrt{N}\left(\mathbb{E}_{\hat{\pi}_N}(f(X)) - \mathbb{E}_\pi(f(X))\right) \sim \mathcal{N}(0, \sigma_{IS}^2(f))$$

where

$$\sigma_{IS}^2(f) = \int \frac{\pi^2(x)}{q(x)}\left(f(x) - \mathbb{E}_\pi(f)\right)^2 dx = Var_q\left(w(X)\left(f(X) - \mathbb{E}_\pi(f)\right)\right)$$

Importance Sampling in High-Dimensions

- Consider the following target distribution

$$\pi(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2}}$$

- We take the following reasonable importance sampling distribution ($\sigma>1$)

$$q_\sigma(\boldsymbol{x}) = \mathcal{N}(0, \sigma^2 I) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2\sigma^2}}$$

- Note that:

$$w_\sigma(x) = \frac{\pi(x)}{q_\sigma(x)} = \sigma^d e^{-\frac{1}{2}\sum_{i=1}^{d} x_i^2 (1-\frac{1}{\sigma^2})} \leq \sigma^d \quad \forall x$$

Importance Sampling in High-Dimensions

$$q_\sigma(\boldsymbol{x}) = \mathcal{N}(0, \sigma^2 I) = \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2\sigma^2}} \qquad w_\sigma(x) = \frac{\pi(x)}{q_\sigma(x)} = \sigma^d e^{-\frac{1}{2}\sum_{i=1}^{d} x_i^2 \left(1-\frac{1}{\sigma^2}\right)} \leq \sigma^d \ \forall x$$

- We now note that:

$$Var_{q_\sigma}\left(\frac{\pi(x)}{q_\sigma(x)}\right) = \int q_\sigma \frac{\pi^2(x)}{q_\sigma^2(x)} dx - \left(\int \pi(x) dx\right)^2 =$$

$$= \int \frac{1}{\left(2\pi\right)^{d/2}} \sigma^d \exp\left\{-\frac{\sum_{i=1}^{d} x_i^2}{2}\left(2 - \frac{1}{\sigma^2}\right)\right\} dx - 1 = \sigma^d \left(\frac{\sigma^2}{2\sigma^2 - 1}\right)^{d/2} - 1 = \underbrace{\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{d/2}}_{>1} - 1$$

- It is easy to see that: $\sigma^4 > 2\sigma^2 - 1 \Leftrightarrow \left(\sigma^2 - 1\right)^2 > 0$. Therefore:

$$Var_{q_\sigma}\left(\frac{\pi(x)}{q_\sigma(x)}\right) \to \infty \ as \ d \to \infty$$

- The variance of the weights increases exponentially fast with dimensionality. This is despite the good choice of $q(x)$.

Introduction to Markov Chain Monte Carlo

- **Markov chain:** A sequence of random variables $\{X_n, n \in \mathbb{N}\}$ defined on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ such that for any $A \in \mathcal{B}(\mathcal{X})$ the following probability condition is satisfied:

$$\mathbb{P}(X_n \in A \mid X_0, \ldots, X_{n-1}) = \mathbb{P}(X_n \in A \mid X_{n-1})$$

and we write:

$$Transition\ Kernel: P(x, A) = \mathbb{P}(X_n \in A \mid X_{n-1})$$

- **Markov Chain Monte Carlo (MCMC):** Given a target distribution π, we need to design a transition kernel P such that asymptotically

$$\frac{1}{N}\sum_{n=1}^{N} f(X_n) \overset{N\to\infty}{\to} \int f(x)\pi(x)dx \ and \ / \ or \ X_n \sim \pi$$

- It is easy to simulate the Markov Chain even if π is complex.

Gibbs Sampler

- If $\theta = (\theta_1, \theta_2, ..., \theta_p)$ where *p>2*, the Gibbs sampler still applies.
- Initialization:
  - Select deterministically or randomly $\theta^{(0)} = (\theta_1^{(0)}, \theta_1^{(0)}, ..., \theta_p^{(0)})$
- Iteration i, i ≥ 1
  - For k=1:p
    - Sample $\theta_k^{(i)} \sim \pi(\theta_k \mid \theta_{-k}^{(i)})$

      where $\theta_{-k}^{(i)} = (\theta_1^{(i)}, ..., \theta_{k-1}^{(i)}, \theta_{k+1}^{(i-1)}, ..., \theta_p^{(i-1)})$

Metropolis -Hastings Algorithm

- Let $\pi(\boldsymbol{x})$ the target and $q(\boldsymbol{y} \mid \boldsymbol{x})$ any (symmetric or not) distribution such $q(\boldsymbol{y} \mid \boldsymbol{x}) = q(\boldsymbol{x} \mid \boldsymbol{y})$. Given state $\boldsymbol{x}_n$ at step n
  - Draw a proposal $\boldsymbol{y}$ *from* $q(\boldsymbol{y} \mid \boldsymbol{x}_n)$
  - Calculate acceptance ratio:

$$\alpha(\boldsymbol{x}_n, \boldsymbol{y}) = \min\left\{1, \frac{\pi(\boldsymbol{y})}{\pi(\boldsymbol{x}_n)} \frac{q(\boldsymbol{x}_n \mid \boldsymbol{y})}{q(\boldsymbol{y} \mid \boldsymbol{x}_n)}\right\}$$

  - Set

$$\boldsymbol{x}_{n+1} = \begin{cases} \boldsymbol{y} & \text{with probability } \alpha(\boldsymbol{x}_n, \boldsymbol{y}) \\ \boldsymbol{x}_n & \text{with probability } 1-\alpha(\boldsymbol{x}_n, \boldsymbol{y}) \end{cases}$$

W. Hastings, Monte Carlo Sampling Methods using Markov Chains and their Applications, Biometrica, Vol. 57(1), pp. 97-109 (1970).

Composition of MH Kernels

- Assume we use a composition of these kernels, then the resulting algorithm proceeds as follows at iteration i.

- **MH Step to Update Component 1**

  - *Sample* $\theta_1^* \sim q_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \cdot\right)$ *and compute*

$$\alpha_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \left(\theta_1^*, \theta_2^{(i-1)}\right)\right) = \min\left(1, \frac{\pi\left(\theta_1^* \mid \theta_2^{(i-1)}\right) q_1\left(\left(\theta_1^*, \theta_2^{(i-1)}\right), \theta_1^{(i-1)}\right)}{\pi\left(\theta_1^{(i-1)} \mid \theta_2^{(i-1)}\right) q_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \theta_1^*\right)}\right)$$

  - *With probability* $\alpha_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \left(\theta_1^*, \theta_2^{(i-1)}\right)\right)$, *set* $\theta_1^{(i)} = \theta_1^*$;

  otherwise set $\theta_1^{(i)} = \theta_1^{(i-1)}$

Composition of MH Kernels

- Assume we use a composition of these kernels, then the resulting algorithm proceeds as follows at iteration i.

- **MH Step to Update Component 2**

  - *Sample* $\theta_2^* \sim q_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \cdot\right)$ *and compute*

$$\alpha_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \left(\theta_1^{(i)}, \theta_2^{*}\right)\right) = \min\left(1, \frac{\pi\left(\theta_2^* \mid \theta_1^{(i)}\right) q_2\left(\left(\theta_1^{(i)}, \theta_2^{*}\right), \theta_2^{(i-1)}\right)}{\pi\left(\theta_2^{(i-1)} \mid \theta_2^{(i)}\right) q_1\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \theta_2^{*}\right)}\right)$$

  - *With probability* $\alpha_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \left(\theta_1^{(i)}, \theta_2^{*}\right)\right)$, *set* $\theta_2^{(i)} = \theta_2^*$;

  otherwise set $\theta_2^{(i)} = \theta_2^{(i-1)}$

Sequential Importance Sampling

At step n=1:

- Sample $X_1^{(i)} \sim q(x_1 \mid y_1), i = 1, \ldots, N$ and then approximate:

$$\hat{p}_N(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1), \; W_1^{(i)}\left(X_1^{(i)}, y_1\right) \propto \frac{\mu\left(X_1^{(i)}\right) g\left(y_1, X_1^{(i)}\right)}{q\left(X_1^{(i)} | y_1\right)}$$

At step n≥2:

- Sample $X_n^{(i)} \sim q\left(x_n \mid y_n, X_{n-1}^{(i)}\right), n = 1, \ldots, N,$ and compute:

$$\hat{p}_N(\boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),$$

$$W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)} | X_{n-1}^{(i)}\right) g\left(y_n | X_n^{(i)}\right)}{q\left(X_n^{(i)} | y_n, X_{n-1}^{(i)}\right)}$$

- The algorithm has computational complexity $\mathcal{O}$(N) independent of n.

Sequential Importance Sampling

- Note that the complexity of the algorithm does not increase with *n*.
- The algorithm is fully parallelizable.
- Also note that if our interest is on computing the marginal posterior,

$\hat{p}_N(x_n|\boldsymbol{y}_{1:n})$ (posterior filtered density), then we only need to store $\boldsymbol{X}_{n-1:n}^{(i)}$ rather than all the $\boldsymbol{X}_{1:n}^{(i)}$ paths

$$\hat{p}_N(x_n|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}(x_n),$$

$$W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)}|X_{n-1}^{(i)}\right) g\left(y_n|X_n^{(i)}\right)}{q\left(X_n^{(i)}|y_n, X_{n-1}^{(i)}\right)}$$

- One can show that this approaches the true posterior as $N \to \infty$.
  - Crisan, D., P. D. Moral, and T. Lyons (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields* 5(3), 293–318.

Conditional Dynamic Linear Model

- A conditional dynamic linear model (CDLM) can be generally defined as

$$Z_n = A(X_n)Z_{n-1} + B(X_n)V_n \quad , \quad Z_1 \sim \mathcal{N}(m_1, \Sigma_1), \quad V_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma_v)$$

$$Y_n = C(X_n)Z_n + D(X_n)W_n \quad , \quad W_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma_w)$$

where all coefficient matrices are functions of the indicator random variable X_n. Both $\{X_n, Z_n\}, n \geq 1$, are unobserved.

- The X_n, which can be either continuous or discrete, is a latent indicator process with certain probabilistic structure. For example:

$$X_1 \sim \mu, \quad X_n \mid X_{n-1} = x \sim f(. \mid x)$$

- Here Z_n is an unobserved Markov process and Y_n is the corresponding observation.
- An important feature of the CDLM is that given the trajectory of the indicator variable X_n, the system is Gaussian and linear.
  - R.Chen,J.S.Liu, Mixture Kalman Filters, Journal of the Royal Statistical Society, 2000,62B:p493

Birth/Death Moves

- Assume a distribution defined on $\{1\}\times\mathbb{R}\cup\{2\}\times\mathbb{R}\times\mathbb{R}$
- We want to propose moves going from (1,θ) to (2, θ_1, θ_2).
- One can propose

$$u \sim g \in \mathbb{R},$$
$$(\theta_1, \theta_2) = h(\theta, u) = (\theta, u)$$

- Its inverse is given by

$$(\theta, u) = h'(\theta_1, \theta_2) = (\theta_1, \theta_2)$$

- The acceptance probability for this birth move is the:

$$\min\left\{1, \frac{\pi(2,\theta_1,\theta_2)}{\pi(1,\theta)}\frac{1}{g(u)}\left|\frac{\partial(\theta_1,\theta_2)}{\partial(\theta,u)}\right|\right\} = \min\left\{1, \frac{\pi(2,\theta_1,\theta_2)}{\pi(1,\theta_1)}\frac{1}{g(\theta_2)}\right\}$$

Latent Variable Viewpoint of EM

- Initialization: Choose initial set of parameters θ^{old}
- E-step: use current parameters θ^{old} to compute $p\left(\mathbf{Z}\middle|\mathbf{X},\theta^{old}\right)$ and then the expected complete-data log-likelihood for general θ

$$Q\left(\theta,\theta^{old}\right)=\sum_{\mathbf{Z}} p\left(\mathbf{Z}\middle|\mathbf{X},\theta^{old}\right)\ln p\left(\mathbf{X},\mathbf{Z}\middle|\theta\right)$$

- M-step: Determine θ^{new} by maximizing $Q\left(\theta,\theta^{old}\right)$

$$\theta^{new}=\arg\max_{\theta} Q\left(\theta,\theta^{old}\right)$$

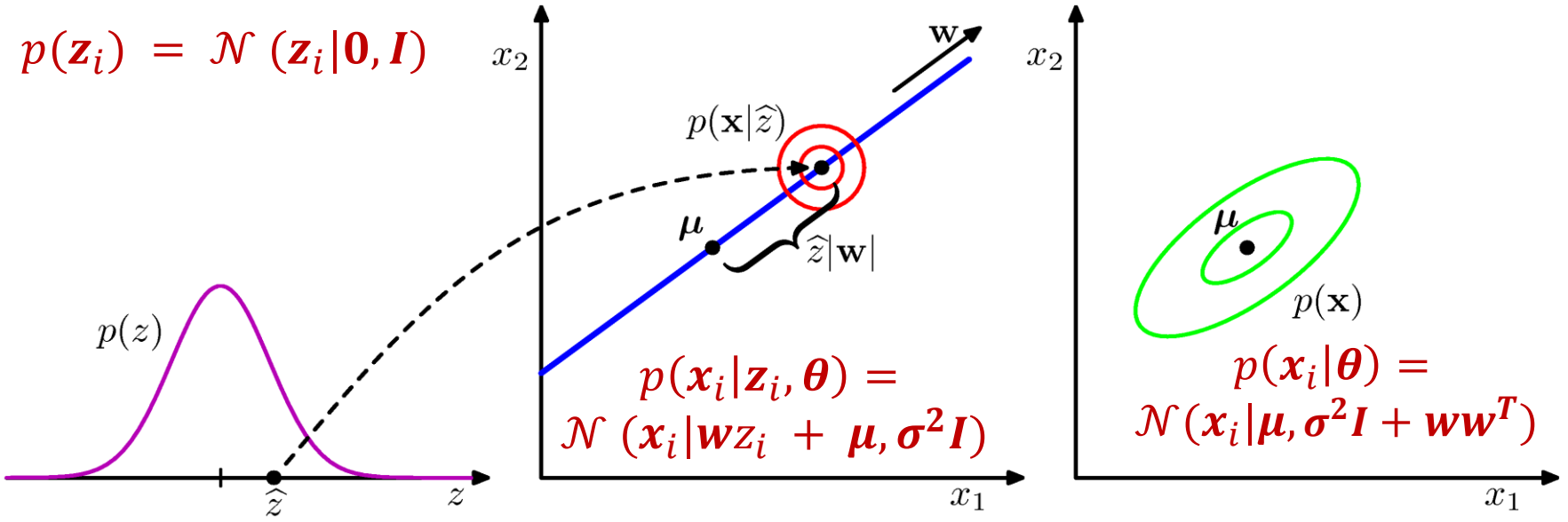
- Check Convergence: stop, or $\theta^{old}\leftarrow\theta^{new}$ and return to E-Step.
- Note: in the definition of $Q\left(\theta,\theta^{old}\right)$ the log acts directly on $p\left(\mathbf{X},\mathbf{Z}\middle|\theta\right)$ *This step is now computationally tractable*.

Generative Point of View

- PPCA is illustrated for $D = 2, M = 1$ (latent dimension).

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \int \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)d\boldsymbol{z}_i = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T)$$

$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I})$



$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ $p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})$ $p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T)$

- Based on the obtained marginal, we see that without sacrificing generality, we can simplify as:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I})$$
$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{W}^T)$$